

CLAIMS

What is claimed is:

1. A queue comprising:
 - a first memory having a first memory access time;
 - 5 a second memory having a second memory access time; and
 - control logic which enqueues in the queue a pointer to data to be transmitted by writing the pointer in the first memory and transferring the pointer to second memory.
- 10 2. A queue as claimed in Claim 1 wherein the first memory access time is less than the second memory access time.
3. A queue as claimed in Claim 1 wherein the control logic transfers a plurality of pointers to the second memory in a single transfer cycle.
- 15 4. A queue as claimed in Claim 1 wherein the control logic enqueues the pointer in the first memory in a single write operation and establishes a linked list of pointers in the second memory after the write operation.
5. A queue as claimed in Claim 1 wherein the control logic dequeues the pointer by reading the pointer from the second memory.
6. A queue as claimed in Claim 1 wherein the control logic fills a cache row in the first memory before transferring the cache row into the second memory.
- 20 7. A queue as claimed in Claim 6 wherein the first memory transfers the cache row in a single write operation.

8. A queue as claimed in Claim 1 wherein the control logic partially fills a cache row in the first memory before transferring the cache row into the second memory in a single write operation.
- 5 9. A queue as claimed in Claim 1 wherein entries in a cache row in the first memory are ordered by position in the cache row.
10. A queue as claimed in Claim 1 wherein the first memory includes two cache rows.
- 10 11. A queue as claimed in Claim 1 wherein a packet vector stored in the second memory includes a cache row entry and a count of the number of pointers stored in a cache row entry.
12. A queue as claimed in Claim 1 or Claim 11 wherein a packet vector stored in the second memory includes a link to a next packet vector in the queue.
- 15 13. A queuing method comprising the steps of:
writing in a first memory having a first memory access time a pointer to data to be transmitted; and
transferring the pointer to a second memory having a second memory access time.
- 20 14. A queuing method as claimed in Claim 13 wherein the first memory access time is less than the second memory access time.
- 25 15. A queuing method as claimed in Claim 13 wherein the step of transferring forwards a plurality of pointers to the second memory in a single transfer cycle.

0955940000427000

Sub
A+

16. A queuing method as claimed in Claim 13 wherein the step of writing writes the pointer in a single write operation to the first memory and establishes a linked list of pointers after the write operation.
17. A queuing method as claimed in Claim 13 further comprising the step of:
5 dequeuing the pointer from the second memory.
18. A queuing method as claimed in Claim 13 wherein the step of transferring forwards a full cache row into the second memory.
19. A queuing method as claimed in Claim 13 wherein the step of transferring forwards a partially filled cache row into the second memory.
- Sub A2 10 20. A queuing method as claimed in Claim ~~18~~ or Claim 19 wherein the cache row is transferred in a single write cycle.
21. A queuing method as claimed in Claim 13 wherein entries in a cache row in first memory are ordered by position in the cache row.
22. A queuing method as claimed in Claim 13 wherein the first memory includes
15 two cache rows.
23. A queuing method as claimed in Claim 13 wherein a packet vector stored in the second memory includes a cache row and a count of the number of pointers stored in the cache row.
- Sub A3 20 24. A queuing method as claimed in Claim ~~13~~ or 23 wherein a packet vector stored in the second memory includes a link to a next packet vector.

25. A queue comprising:
a first memory having a first memory access time;
a second memory having a second memory access time; and
means for controlling the queue enqueues in the queue a pointer to data
to be transmitted by writing the pointer in the first memory and transferring the
pointer to second memory.
26. A queue as claimed in Claim 25 wherein the first memory access time is less
than the second memory access time.
27. A queue as claimed in Claim 25 wherein the means for controlling the queue
transfers a plurality of pointers to second memory.
28. A queue as claimed in Claim 27 wherein the plurality of pointers are transferred
in a single transfer cycle.
29. A queue as claimed in Claim 27 wherein the means for controlling the queue
enqueues the pointer in a single write operation to the first memory and
establishes a linked list of pointers after the write operation.
30. A queue as claimed in Claim 27 wherein the means for controlling the queue
dequeues the pointer by reading the pointer from the second memory.
31. A queue as claimed in Claim 27 wherein the means for controlling the queue
fills a cache row in the first memory before transferring the cache row into the
second memory.
32. A queue as claimed in Claim 31 wherein the cache row is transferred to the
second memory in a single write operation.

33. A queue as claimed in Claim 25 wherein the means for controlling the queue partially fills a cache row in the first memory before transferring the cache row into the second memory.

5 34. A queue as claimed in Claim 25 wherein entries in a cache row in the first memory are ordered by position in the cache row.

35. A queue as claimed in Claim 25 wherein the first memory includes two cache rows.

10 36. A queue as claimed in Claim 25 wherein a packet vector stored in the second memory includes a cache row entry and a count of the number of pointers stored in a cache row entry.

37. A queue as claimed in Claim 25 or 36 wherein a packet vector stored in the second memory includes a link to a next packet vector in the queue.

15 ~~38.~~ A pointer list comprising:
a first memory having a first memory access time;
a second memory having a second memory access time; and
control logic which adds in the pointer list a pointer to data to be
transmitted by writing the pointer in the first memory and transferring the
pointer to second memory and removes the pointer from the pointer list by
20 reading the pointer from the second memory.

Add
AS 7